SICS/T-89/8908

A Learning System for Fault Finding

by

Anders Tunevi

SICS technical report
T89008
ISSN 1100-3154

A Learning System for Fault Finding

Anders Tunevi

Infologics AB
Box 91
191 22 Sollentuna
SWEDEN
Tel: +46 8 92 20 00

April 5, 1989

Abstract

A learning system for fault finding has been constructed. This system contains many different types of knowledge, three ways to find faults and four ways to learn fault finding.The constructed learning system works for a class of fault finding problems.This class has been described in the paper. The developed system could be viewed as an architecture of a general learning system for fault finding.The system could also be used as a testbench of learning mechanisms.

The experiences from this project indicates that it is possible to build a learning system when the structure of the knowledge is known.

In this paper the following ideas will be discussed: How can the fault finding and learning techniques be integrated? How can the knowledge structure, fault finding mechanisms and learning mechanisms emerge with the help of a simulator and general mechanisms?

A learning system for fault finding

1. Introduction

The **main purpose** with this project is to explore how "learning" can be used in fault finding.The **plan** was to select one object (a Sesam terminal) as an example and carry out the following activities:

> Find knowledge associated with the troubleshooting of a Sesam terminal. Study different types of knowledge as: heuristic rules, the functional and the physical description of a Sesam terminal, how the different types of knowledge are related to each other. Discuss criteria for deciding what types of knowledge should be saved in the knowledgebase and what should be derivable. Build a knowledgebase containing different types of knowledge.
>
> Find different ways to find faults in a Sesam terminal using different types of knowledge. Make models of some of the different ways of finding faults.
>
> Find different ways to learn how to troubleshoot a Sesam terminal. Make models of the different ways of learning.

While studying and constructing the models, the following questions were discussed: What would happen if the complete Sesam system instead of just one component (a Sesam terminal) was selected? How general was the model ? Could it be extended to a general mechanism? What were the problems? How could the different techniques be integrated?

1.1 Background

The background of the project consists of experiences from the construction of a fault finding system for a computer called Sesam, studies of different learning techniques and studies of different systems for fault finding.

Sesam is a system that helps telephone operators in about 150 companies in Sweden. It is an appendage to a company's switchboard consisting of a control unit and one terminal per telephone operator. The Sesam Fault Finder instructs a service technician how to find and repair a fault in a Sesam computer. The system runs on a portable PC. The construction of the Sesam fault finder (Sesam FF) was initiated by the Swedish Telecom. The project team consisted of one knowledge engineer and the constructors of the Sesam computer.

Building an expert system for fault finding was a time-consuming process. Developing a learning system for fault finding could be one way to change this. The idea of using learning in fault finding emerged from the paper "Learning in second generation expert systems " [8]. This paper describes a system that uses an ordinary car as a source for a concrete example of a learning system for fault finding.

To be able to test and experiment with different learning techniques a simulated "world", consisting of a Sesam terminal, an expert and a novice was constructed.

In the paper "Acquisition of Proof Skills in Geometry" [1] a model of how students learn to solve geometry problems is presented. This paper gave me the idea of using

my own experiences of how I learned how to troubleshoot a Sesam terminal to develop a learning system.

To build a learning system for fault finding I had to know what should be learned and what fault finding mechanisms should be used. In the Sesam FF project I had studied how technicians found faults in Sesam and also learned how to repair Sesam myself. I also studied different papers about fault finding as "Expert Systems Technology" [2] and "A theory of diagnosis from first principles" [6].

Learning can be viewed as a process where pieces of knowledge continuously are changing. Acquiring a list of these pieces of changing knowledge generates the steps of the learning process.I started to build a list of different types of knowledge associated with the fault finding of a Sesam terminal.Some types of knowledge from this list were selected and put into a knowledge base. Fault finding mechanisms were constructed.

I learned fault finding of Sesam in different ways. I was given instructions how to do, I experimented with the terminal and so on. The ideas of what learning techniques to use emerged from these experiences.There exist other projects with the idea of using many learning techniques. Disciple [7] is an example of a system that integrates different learning techniques.

The ideas of this project are similar to the ideas in the papers presented above. But I have not found any project based on the following ideas:

- trying to make a list of all knowledge in one domain
- using an architecture consisting of a simulated expert,a simulated novice and a simulated object
- using learning by questioning in the field of fault finding

2 The system: An overview

A picture of the system is showed in figure 1.The system consists of three main components: A simulated Sesam terminal, a simulated expert troubleshooter of a Sesam terminal and a simulated novice

The Sesam terminal is simulated by a forward chaining mechanism interpreting the deep knowledge.The simulated expert troubleshooter produces a trace while troubleshooting the Sesam terminal and is able to answer questions from the simulated novice. The simulated novice is supposed to learn how to troubleshoot the Sesam terminal in different ways.

3 Knowledge structures in fault finding

A list of knowledge associated with the fault finding of a Sesam terminal

A list of different types of knowledge associated with faults in a Sesam terminal have been produced. Here is an overview of the knowledge found: knowledge about the

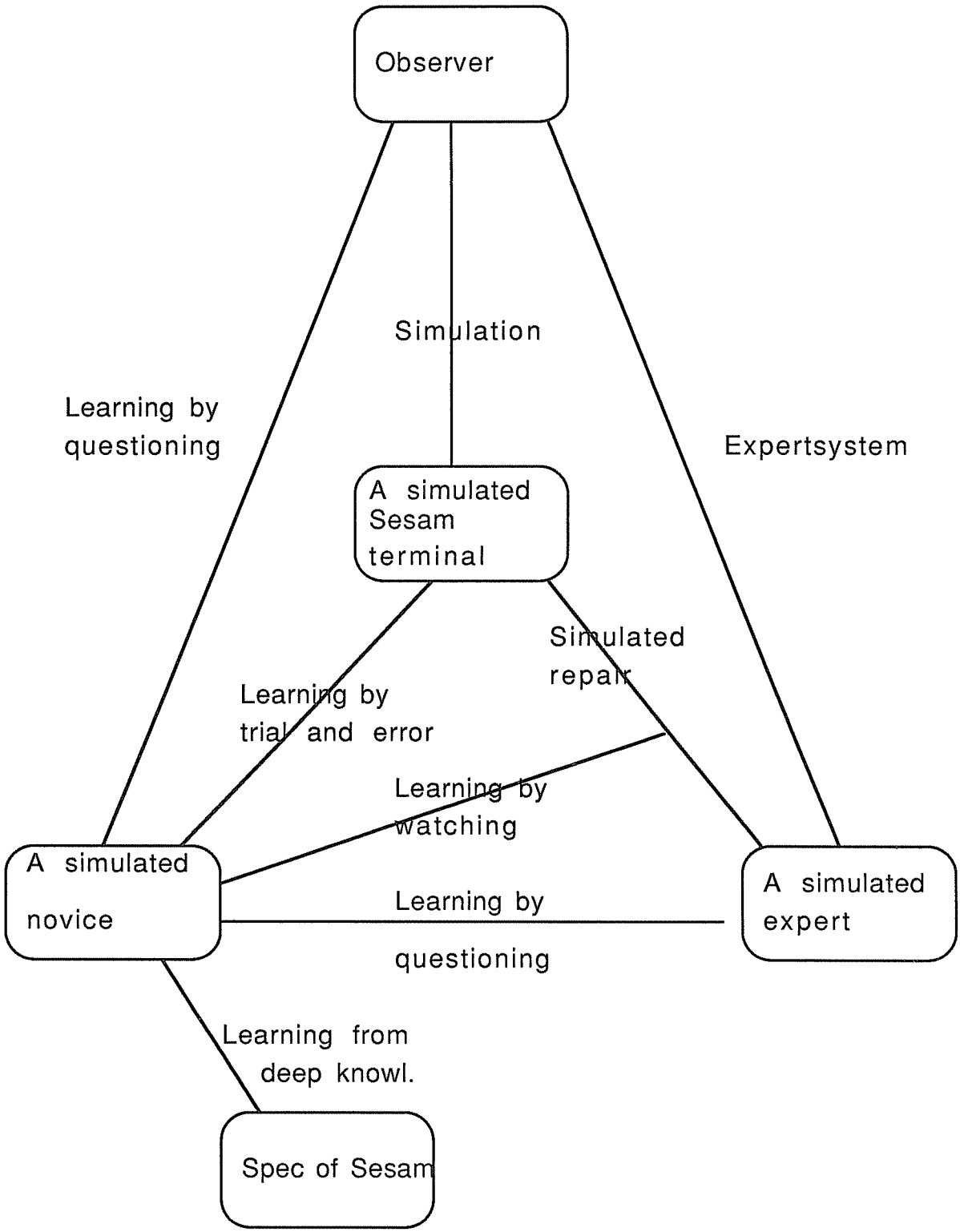

Figure 1 A learning system

physical structure, knowledge about faults,knowledge about the functional structure, primitive actions and results, knowledge about tests, troubleshooting knowledge, common sense knowledge,relations between different types of knowledge and case knowledge.

3.1 Background knowledge

Here is a list consisting of suitable assumptions in the domain of fault finding a Sesam terminal. The learning and fault finding mechanisms that will be presented later are appropriate for the class of fault finding problems that are characterized by this list.

1. The object consists of parts.
2. Some of the parts are connected.
3. Every part is linked to an action: repair
4. Repairing a faulty part makes the part function properly.
5. Every connection is linked to three actions: disconnect, connect and check
6. Checking or connecting a connection makes the connection correct.
7. The whole object is functioning properly if all connections and all parts belonging to the object are functioning properly.
8. A fault can not cause another fault. Faults does not emerge while fault finding.
9. There exist tests and every test has one result indicating that the subsystem is correct and one or more results indicating a fault in the subsystem.
10. Every test is linked to a question.
11. Some of the tests need different configurations.
12. One of the tests is the main test. The main test is used to test whether the object is functioning properly.
13. Every test with a value indicating a fault has a set of possible causes.
The relation between, two sets of causes of results indicating faults, belonging to two different tests is one of the following:
    A.The sets are disjoint.
    B. One of the sets is a subset of the other set.
14. The tests can be ordered according to simplicity.For every test with a result indicating a fault there exist a list. This list consists of actions and subtests. The subtests are all simpler than the test. If all subtests in the list are ok and all actions are performed then the test will no longer give a result indicating a fault.

A knowledgebase was constructed from parts of the knowledge found in the investigation of knowledge described.Here are some examples of knowledge from the knowledgebase: actionlists, rules, knowledge about tests, knowledge about the physical structure and knowledge about the functional structure

3.2 Two main groups of knowledge

The knowledgebase could be divided into the following two main groups of knowledge: deep knowledge and shallow knowledge. The shallow knowledge consists of rules/actionlists, knowledge about tests and knowledge about the physical structure. The deep knowledge consists of knowledge about the functional structure and the physical structure.

It is possible to partly derive the shallow knowledge from the deep knowledge. It is only possible to derive the shallow knowledge partly, because the deep knowledge does not contain any knowledge about how difficult/easy it is to make the different actions or how often a certain fault occurs.

Actionlist is a type of knowledge that is used a great deal in this work and will therefore be explained. An actionlist consists of: a test with a faulty value and an ordered list of actions and subtests.Here is an example of an actionlist together with an explanation:

actionlist([quicktest,screen_black], [[reset_screen_test,screen_black], [repair,control–unit]]).
Explanation : If quicktest gives as a result that the screen is black then try to make quicktest correct by making one of the following: making the reset_screen_test correct or repairing the control unit

4 Strategies in fault finding

The way a technician find faults could depend on his competence.Eg:A technician experienced with a Sesam terminal could work from rules telling him directly how to find a fault. A technician with knowledge of how a Sesam terminal functions uses this knowledge to find the fault.

The way a technician find faults could also depend on the actual assumptions.When one troubleshoots an object one could use different assumptions such as: there exists just one fault, one fault cannot cause another fault, the things one uses to exchange the different parts are correct. If one does not manage to find the cause of some symptom one drops some of the assumptions

Different methods for fault finding have been developed.Two methods for the simulated expert to find a fault in a Sesam terminal have been implemented. The two methods use the following different types of knowledge: shallow knowledge and deep knowledge.

One method for the simulated novice to find a fault in a Sesam terminal has been implemented. This method is a combination of two methods one working from rules and the other working from knowledge about the physical structure. This method is used in the learning by "trial and error" routine.

4.1 Fault finding using shallow knowledge

A model of how a technician could find a fault in a Sesam terminal using shallow knowledge is implemented in the form of production-rules with a forward-chaining mechanism. The inference mechanism together with a small set of rules forms a mechanism for fault finding. This mechanism works on actionlists and knowledge about tests. The algorithm is as follows:

Put a symptom on top of the problemstack and continue with the following activities until the problemstack is empty:

1. If the element on top of the problemstack is a test then ask of the result of the test and proceed as follows:
    A. If the result of the test indicates correctness then take away the test from the problemstack.
    B. If the result of the test indicates a fault then use the actionlist for the test to find a subtest or an action that has not been tried. Put this action/subtest on top of the problemstack.
2. If the first element in the problemstack is an action then make this action and take away this action from the problemstack.

4.2 Fault finding using deep knowledge

A model of how faults could be found using the theory of Reymond Reiter [6] has been constructed . This method uses a description of the system (the functional description) and some observations. The algorithm gives a list of possible faults as a result. This list consists of lists that are minimal subset of faulty components that explains the observations.The algorithm is as follows:

1. Make a node that consists of:
- assumed faulty components = empty
- rest-components = consists of all components
- type = expandable
2. Select nodes with the type "expandable" until there are no more such nodes. For every such node proceed as follows:
   A Check if the assumed faulty components are consistent with the observation. Use the functional description to make this check.
   B. If the assumed faulty components are consistent with the observations then mark this node with ok. Take away surplus nodes.
   C. If the assumed faulty components are not consistent with the observations expand this node by doing the following: Move one component from rest-components to assumed faulty components in every possible way. Make a new node in every such case if the generated node is not covered by any existing node. Mark the generated nodes with type = expandable.
3. Collect all nodes that have been marked with ok and present their assumed faulty components as the result.

5 Learning processes in fault finding

The learning process could be divided into two phases: The collecting of knowledge and the transforming of knowledge into other types of knowledge. How one learns fault finding of a Sesam terminal depends on the source of knowledge. The following sources have been found: an expert answering questions, an expert solving problems, a description of a Sesam terminal and the Sesam terminal itself. One way of learning has been developed for each source. A picture of the different ways of learning is showed in figure 2. "Learning from deep knowledge" uses the description of the Sesam-terminal. "Learning by trial and error" uses the Sesam terminal itself. "Learning by questioning" uses an expert answering questions. "Learning by watching" uses an expert solving problems.

Two of the methods ("learning from deep knowledge" and "learning by watching") use deduction to transform the input knowledge into actionlists. One of the methods "learning by trial and error" uses induction and one "learning by questioning" is interactive.

5.1 Learning by questioning

The model of how to learn using questioning works in an interactive way. The knowledge used: The answers of the expert. Knowledge produced : Physical structure, actionlist, knowledge about tests and knowledge about actions (eg how to repair parts). The mechanism is described below.

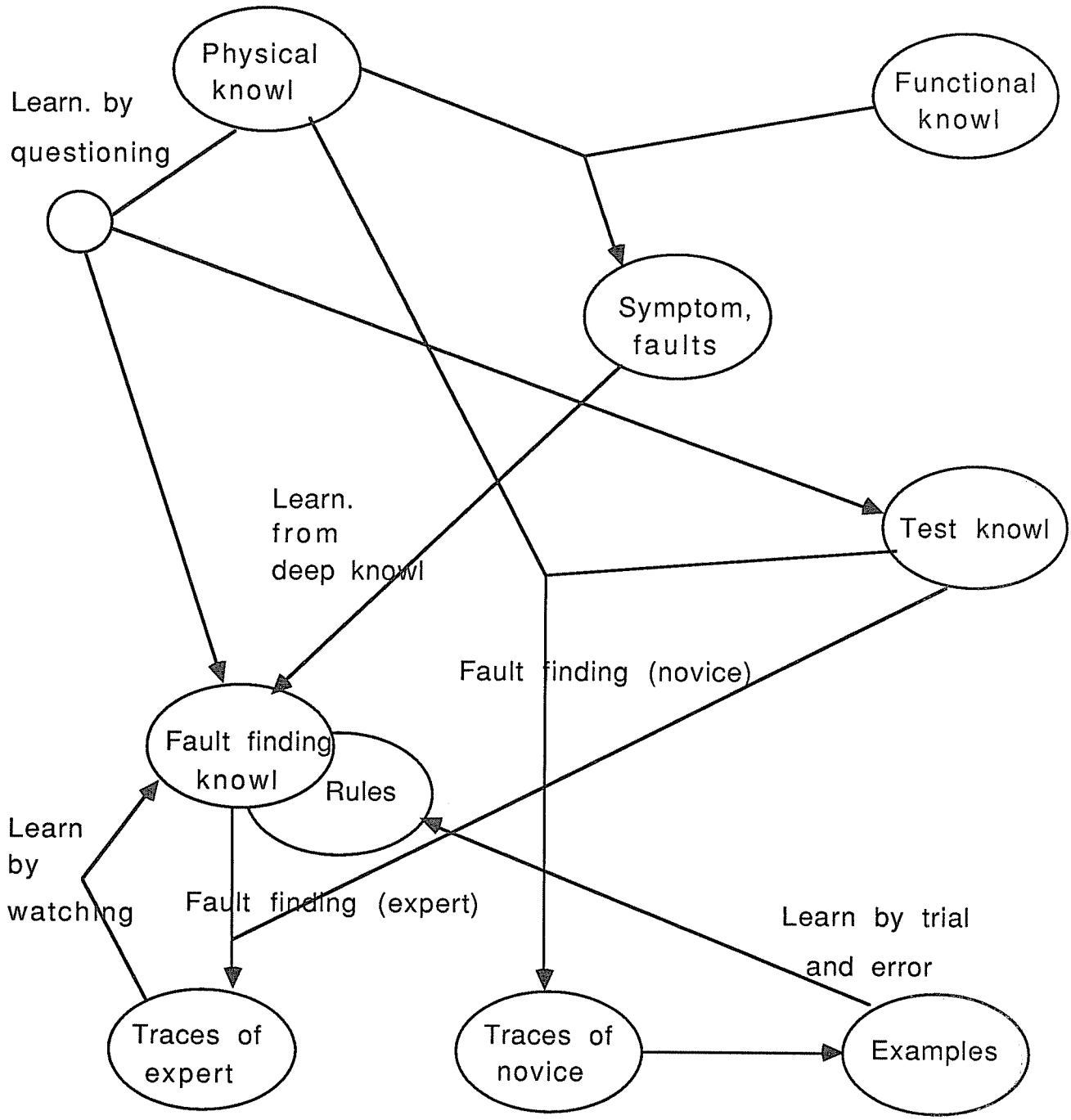

Figure 2 Learning

1. The system asks about the physical structure:

- What parts the object consists of.
- How these parts are connected.

2. The system asks about tests:
- What tests exist.

3. For every test the system asks about the following:
- A description of how the test is made.
- A list of possible results of the test.
- A list of connections that the test uses. (the configuration may vary depending on different tests)

4. For each pair consisting of a test and a result of the test that shows a fault in the object the system asks for a list of subtests/actions.
5. The system asks for a description of how to repair every part in the object.
6. The system asks for a description of how to connect, disconnect and check every connection in the object.

The order of the questions is important. The knowledge acquired by one question is used to form the next question(s).

5.2 Learning from deep knowledge

In the model of how to learn using deep knowledge actionlists are produced from knowledge about the physical structure, the faults and the functional structure. The algorithm is as follows:

1. Every possible fault is simulated. For every possible fault every test is tried. For every pair consisting of a test and faulty value all possible causes (= faults) are collected into a list.
2. Put the main test in a que of tests and proceed as follows until the que of tests is empty: Chose a test from the que and for every pair consisting of a test and a result indicating a fault make the following:
   A. Use the lists produced in 1 and make the following:Divide the test and the faulty value into subtests as far as possible. Put subtests in the que.
   B. Make actions of the causes of the test that are not covered by any subtest.
   C. Make an actionlist consisting of the test and the result as first argument and a list consisting of the subtests produced in A and the actions produced in B as second argument.

When we compared the knowledge produced from this module with the knowledge produced from the routine "learning by questioning" differences were found. When the differences were investigated there were faults found in both the functional description and in the action lists.This shows that the use of techniques using different input knowledge to acquire the same output knowledge could be a way of cross-checking the knowledge.

5.3 Learning by "watching" the expert working

A model of how to learn watching the expert will now be presented.Traces from the simulated expert are converted to actionlists. The system could be viewed as the inverse of the fault finding mechanism.The background knowledge of this mechanism consists of: the general background knowledge described and the assumption that the simulated expert finds fault using the fault finding mechanism described. The algorithm is as follows:

Proceed as follows until the trace is empty:

1.If the first element of the trace is a test with a result indicating a fault then:
A. Make an actionlist for this test using the trace.
B. Take away this test from the trace and make actionlists of the rest of the trace.
2. If the first element of the trace is not a test with a result indicating a fault then take away the first element of the trace and make actionlists of the rest of the trace.

The actionlist produced from the trace is combined with the actionlists produced from other traces.

5.4 Learning by trial and error

A model of how to learn using "trial and error" will now be presented. The following knowledge is used: possible faults, knowledge about the physical structure, list of tests and descriptions of tests. The following knowledge is produced using induction: traces gives examples and examples gives rules. The general background knowledge described with the exception that there are no conditions on the tests (point 13 and 14 in the background knowledge) constitutes the background of the mechanism.The system works in the following cycle:

1. Generate a fault.
2. If the system solves the problem with the existing rules go back to 1.
3. Try all tests.
4. Try all possible actions until the main-test is ok.
5. Generate an example from the trace and add this example to the collected examples.
6. Induce new rules from the collected examples.AQ15 [5] is used.
7. Go back to 1.

6 Discussion

It existed a lot of different types of knowledge that could be associated with the fault finding of a Sesam terminal. It would not be suitable to save all. How to decide what types of knowledge to put into a knowledgebase? An example of a question that has risen is: Should one allow redundancies in the knowledgebase? If two representations are derivable from each other should one save both or just one? How should one do with concepts that are similar but not exactly the same? Eg: connected to versus connectable. A framework to decide the right combination of pieces of knowledge to put into the knowledgebase is probably needed.

Different types of fault finding problems seem to need different fault finding mechanisms working from different types of knowledge.

The learning mechanisms are appropriate in different situations. More learning mechanisms could have been produced by combining the input knowledge of one mechanism with the method of another mechanism.

The constructed learning system works for a class of fault finding problems (see background knowledge) that have been described in the paper. The developed system could also be viewed as an architecture of a general learning system for fault finding.The system could also be used as a testbench of learning mechanisms.

The experiences from this project indicates that it is possible to build a learning system when the structure of the knowledge is known.

6.1 Future work

Two project ideas for future work will be presented. The purpose with this the **first project idea** is to investigate the possibility of extending the developed prototype to use it on real cases.The plan is:

1.Select a more complicated object. (eg the complete Sesam). Adjust the knowledgebase and mechanisms to the class of fault finding problems that this object belongs to.
2.Build an expert interface to make it easy to add knowledge.
3. Integrate the developed learning techniques. This should make it possible to learn from different sources of knowledge where the knowledge may be inconsistent and incomplete.
4. Test the developed system on a new case.

The purpose with the **second project idea** is to investigate if it is possible to build a learning system that is able to invent new types of knowledge representations, learning and fault finding mechanisms for different classes of fault finding problems. This project could be viewed as a theoretical continuation of the work described and the idea has emerged from the AM and EURISCO project described in [3] and [4].The plan is:

1.Investigate how different types of knowledge associated with the fault finding of a Sesam terminal could be invented. Use the list of different types of knowledge and investigate how they are related. Does there exist a path from a small set of knowledge to more and more complicated knowledge?
2. Investigate how the mechanisms in the developed system could be generated.(eg from the background knowledge)
3. Let the simulated novice experiment with the simulator of a Sesam terminal working only from a small set of knowledge. Investigate if it is possible for him to learn the knowledge types, background knowledge, fault finding and learning mechanisms. Only general mechanisms should be used. Test the mechanisms developed on another example.

7 References

[1] Andersson J. R.: Acquisition of Proof Skills in Geometry
Michalskie R.S, Carbonell J.G., Mitchell,T.M..: Machine Learning
An Artificial Intelligence Approach 1984

[2] Keravnou ET.,Johnson L.: Expert Systems Technology 1985

[3] Lenat D.B. :Nature of heuristics. Theory Formation by Heuristic Search.
Pearl J.:Search and Heuristics, 1983

[4] Lenat D.B. :The role of heuristics in learning by discovery: Three case studies
Michalskie R.S, Carbonell J.G., Mitchell,T.M..: Machine Learning
An Artificial Intelligence Approach 1984

[5] Michalski R.S., Mosetic I., Hong J., Lavrac N.: The AQ15 Inductive Learning System: An Overview and Experiments by July 1986

[6] Reiter. R:A theory of diagnosis from first principles. Artificial Intelligence Volume 32 Number 1 April 1987

[7] Tecuci G.:Disciple: A Theory , Methodology and System for Learning Expert Knowledge.

[8] Van De Velde L. S. W.: Learning in second generation expert systems .
This article apperaed in J.S. Kowalik 1985